

Node package manager (NPM)

Get all the details of package

```
npm view <package_name>
```

Install Package

```
npm install <package_name>
```

```
// OR
```

```
npm i <package_name>
```

Remove / Uninstall Package

```
npm remove <package_name>
```

```
//OR
```

```
npm uninstall <package_name>
```

Install Package Globally

This will install specified package in your system (globally), not in your working project folder

```
npm i <package_name> -g
```

```
// OR
```

```
npm i <package_name> --global
```

Get the list of global packages

```
npm -g list
```

```
npm root -g
```

Semantic Versioning

MAJOR.MINOR.PATCH

12.2.1

Major - version when you make incompatible API changes

Minor - version when you add functionality in backwards compatible manner

Patch - version when you make backwards compatible bug fixes

Dependencies

1. Normal dependencies
2. Dev dependencies
3. Peer dependencies

1. Normal Dependencies

```
npm install lodash@latest
```

When you run following command, this will edit package.json file and add lodash as dependency

```
{
  "name": "npm_basics",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "dependencies": {
    "lodash": "^4.17.21"
  }
}
```

We generally don't add **node_modules** folder in our code base, when we want to run our application just give the command

npm install

This will install all dependencies listed in package.json file in any environment

2. Dev-dependencies

These dependencies are used to speed-up our development process like **nodemon**

```
npm install nodemon --save-dev
```

Content of package.json

```
{
  "name": "npm_basics",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "dependencies": {
    "lodash": "^4.17.21"
  },
  "devDependencies": {
    "nodemon": "^2.0.15"
  }
}
```

3. Peer dependencies

Used by developers who are developing npm packages.

NPM-Scripts

<https://docs.npmjs.com/cli/v8/using-npm/scripts>

```
{
  "name": "npm_basics",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "start": "echo \"This script is started\"",
    "custom_script": "echo \"This is custom script\""
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "dependencies": {
    "lodash": "^4.17.21"
  },
  "devDependencies": {
    "nodemon": "^2.0.15"
  }
}
```

To run npm scripts

npm start // no need to specify run for "start" script

npm run custom_script

NPX - Node Package Execute

Allow us to run **npm** tools **without installing them globally**

Allow to run local binaries

Allow to run global binaries without installing them

Examples

This will initialize fresh react app

```
npx create-react-app
```

If you haven't downloaded cowsay package, it will install it then, execut it directly by npx and remove automatically.

```
npx cowsay "Hello how are you?"
```

```
< Hello how are you? >
```

```
-----
```

```
\      ^__^
 \    (oo)\_______
      (__)\       )\/\
         ||----w |
         ||     ||
```